Fault Tolerant Systems Analysis: Dynamic Combinatorial Models

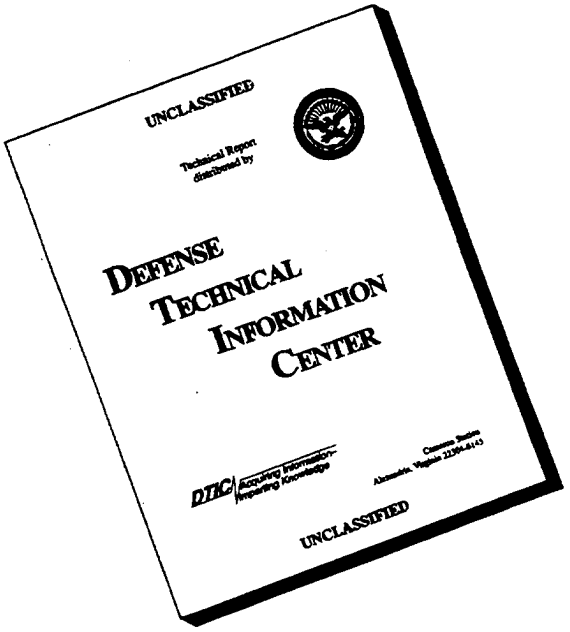
Final Report for grant N00014-88-K-0368

Joanne Bechta Dugan
Department of Computer Science
Duke University
Durham, NC 27706

June 30, 1989

Contents

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

1 Introduction

The problems that arise during reliability analysis of a fault tolerant computer system can be broadly classified into those relating to the construction of the model, and those relating to the solution of the model. The construction of a model of a complex fault tolerant system consists of selecting an appropriate "language" for the description of the system, abstracting the important characteristics of the system to be studied, and expressing these characteristics in the description language. The underlying stochastic representation of the system can then be automatically determined from the description language; the solution of the underlying stochastic process provides estimates of the desired measures. Some examples of modeling languages that are appropriate for simplifying the model construction task are combinatorial models, such as reliability block diagrams [20] and fault trees [2]. Such combinatorial models are useful because they provide a concise representation of the system; however, they are not able to model the dynamic system behavior in response to a fault or an error.

The first topic considered under the auspices of this grant was concerned with the development of techniques for incorporating fault and error modeling techniques into combinatorial models. A second area of research conducted under the current contract concerns the development of fast, accurate algorithms for the solution of fault tree models. Several different techniques were developed for producing bounded approximations for both static and dynamic combinatorial models. (The techniques were applied specifically to fault trees, but are also applicable to reliability block diagrams.) Techniques for the consideration of truncated fault trees were derived which could be used to produce bounded estimates of system reliability from partially developed fault trees.

Other topics considered include the analysis of phased missions; a new technique for combining models for multiple phases was derived. We also investigated the problem of sequence dependencies. Three different types of sequence dependencies were defined, and associated solution techniques were developed. Papers describing the results of these two efforts are in preparation, and will be submitted as they are completed.

2 Solution of Dynamic Combinatorial Models

2.1 Background

Combinatorial models are used for reliability prediction because they provide a concise representation of the failure (or success) modes of a fault tolerant system, but they cannot accurately model dynamic behavior intrinsic to the fault recovery process. For example, they cannot be used to model transient faults, and they cannot address the issues concerned with the amount of time needed to recover from a fault, or even the probability that recovery from a fault is successful. But if the combinatorial model of a system is used in conjunction with the behavioral decomposition technique, the modeler can use a combinatorial model to describe the system structure and fault occurrence behavior, and use a different model type for describing the fault/error handling behavior. One way in which the two may then be combined is by converting the combinatorial model to a Markov chain (which is capable of modeling more complex and dynamic behavior, but which is more cumbersome for the modeler to use), and then adding the dynamic fault/error handling behavior to the resulting Markov chain [6, 11].

The combinatorial model to Markov chain conversion consists of enumerating the sequences of events that may occur. Starting from the initial configuration, each operational configuration becomes a state in the Markov chain. The potential size of the state space is reduced by allowing the combination of identical components into single basic events. This method is applicable to fault trees with replicated inputs, or with complex gates, such as *m-out-of-n* gates with non-identical inputs; it is also applicable to non-series/parallel reliability block diagrams, and to other combinatorial models of non-repairable systems as well. The resulting set of states and transitions constitutes a continuous-time Markov chain (CTMC) [7] which is solved for time dependent state probabilities. The sum of the operational state probabilities gives the reliability of the system. We note that the traditional methods for combinatorial model solution [3, 14, 18, 19, 20, 21] are not applicable once we add the dynamic fault/error-handling behavior.

We recently developed a new algorithm [9] that enables the inclusion of the fault/error handling behavior in the combinatorial model, but which obviates the need for the Markov chain solution. As the state space is expanded in a breadth-first search, each state's contribution to the system unreliability is calculated exactly. This method is applicable to the same class of systems as the conversion to Markov chain approach described in the previous paragraph (fault trees with replicated (shared) inputs, multiple occurrences of identical component types, and complex gates, and to non-series/parallel reliability block diagrams or other combinatorial models of non-repairable systems). The algorithm evaluates the probabilities of single-point failure, near-coincident faults, and exhaustion of redundancy separately. There is no need to store the entire state space of the system; once the leakage from a state is generated, and its children states are generated, the state is discarded.

This new method for combinatorial model solution effectively removes many of the shortcomings of the model. While using a language that is familiar to many reliability engineers, this method (when combined with behavioral decomposition and coverage modeling) allows a more realistic analysis of system unreliability.

2.2 Achievements

Under the auspices of the current contract progress on the development of the algorithm produced several tangible results. First, a prototype version of the algorithm has been implemented, thus demonstrating the feasibility of the approach. This implementation was effected in standard FORTRAN-77 and currently runs on a Sun workstation. Models of several real systems were solved

using the new method, and the solution was compared to more traditional Markov chain solution techniques. In the preliminary version, the new solution method took approximately 50 percent longer than the alternative method, but resulted in the saving of over 1 Mbyte of disk space for intermediate files. The application of this algorithm to analysis of the I/O network of the AIPS (Advanced Information Processing System, under development at C.S. Draper Lab) [, was reported in [9].

Second, a technique for intelligently truncating the model was developed. Since the model construction and solution phases are combined, the model can be truncated at any point in the process, and bounds on the reliability of the system can be produced. If the bounds are determined to be too loose, the solution can continue with no loss of work. This method may be contrasted with other methods [1, 5, 4, 15] in which either the entire model is generated *a-priori* and then reduced, or the model is regenerated from the start when the bounds are unacceptable large. This method also removes from the modeler the burden of trying to guess at which point the truncation should be performed. The modeler can instead define a maximum width of the error interval (as a percentage of the nominal value) that is acceptable, and the solution can proceed until this criteria is met. A paper describing this work [9] is scheduled to appear in the *IEEE Transactions on Reliability* in June 1989.

3 Reduced Fault Tree Models

Fault tree models are well accepted and solution methods are well known [16, 17], but exact analysis of fault trees with many basic events is often quite expensive, both in terms of developing the model and in solving the model once it is developed. We developed several techniques that can be used to reduce the effort associated with developing and solving a fault tree model for a given system. The techniques addressed three different kinds of problems. First, we developed two different techniques for truncating standard solution techniques for static combinatorial models. The first technique applies to systems solved by exhaustive state enumeration, while the second applies to sum of disjoint products techniques. When considered in combination with techniques for truncating the solution of dynamic combinatorial models, one can produce fast, accurate approximations to the reliability of complex fault tolerant systems.

We also developed two methods for reducing the actual fault tree model of a system (as contrasted with the aforementioned techniques for truncating the solution of the fault tree model). We defined a *truncated fault tree* which is an *a-priori* fault tree reduction, and considered techniques for extracting a truncated fault tree from a full fault tree model of a system that may be too large for exact analysis. We then defined an additional gate type *the functional dependency gate* that can help simplify the fault tree description of a complex system.

A paper describing this work [12] will be presented at the *8th Symposium on Reliable Distributed Systems* and will be published in the proceedings.

4 Phased Missions

Fault tolerant systems are often used in missions that are characterized by several phases, where the system structure, failure and recovery processes, or success criteria can change with each phase. The analysis of the reliability of such systems is hampered by the existence of more than one phase, since separate models must be developed for each phase. The problem arises because the models for the separate phases must be linked together, so that the solution at the end of one phase becomes the initial conditions for the beginning of the second phase.

We developed a methodology for automated analysis of phased missions, based on a Markov chain solution. Assuming that the phase change times are deterministic, we presented a methodology for combining models for each phase into one. This results in a model that can be substantially smaller than what is required by other methods. We defined a unified framework for defining the separate phases using fault trees, and for constructing and solving the resulting Markov model. A paper describing this work is being prepared for submission to *IEEE Transactions on Reliability*.

5 Fault Trees and Sequence Dependencies

A major disadvantage of fault tree analysis is the inability of fault tree models to capture sequence dependencies in the system, and still allow an analytic solution. Systems exhibiting various types of sequence dependencies are usually modeled via Markov models. Markov models have the advantage of providing the flexibility to model a very large class of systems, but have the disadvantage of being difficult to construct.

The gap between fault trees and Markov chains can be narrowed by describing as much of the system as possible in terms of a fault trees, converting the fault tree (automatically) to a Markov chain and then altering the Markov chain to reflect the behavior that cannot be captured in the fault tree model. This approach has been used successfully in HARP, where the redundancy management and fault and error handling behavior of the system is automatically incorporated into the Markov chain that is constructed from the fault tree description.

There are several different kinds of sequence dependencies in fault tolerant systems. We identified three such dependencies, and described the definition, implementation and application of specific gates to express these behaviors in fault tree models. The use of these gate types still allow an analytic solution, and are useful in modeling complex fault tolerant systems.

The first type of sequence dependency we described is termed *functional dependency* and it arises in the following situation. Suppose that the failure of some component A causes components B and C to become inaccessible or otherwise unusable, so that they should also be considered to have failed. That is, components B and C are functionally dependent on component A. However, the failure of either B or C does not affect the functionality of A. We present a functional dependency gate to model this situation.

The second type of sequence dependency investigated arises from the use of cold spares (spares that can not fail until switched into active operation). If component B is a cold spare for component A, then the failure of component B cannot occur until after A has failed and B is switched into active operation. We introduced a *cold spare gate* to model this situation, and developed an analytic solution of the resulting model.

The third type of sequence dependency considered occurs in the following situation. Suppose that components A and B have both failed, but system failure occurs *only* if A failed *before* B did. If B failed before A did, then the system is still operational. We defined an implementation of the *Priority AND* gate [13][16] to model this situation. Our implementation allows an exact solution, and is comparable to Fussel's solution. A paper describing this work [10] will be presented at *1990 Reliability and Maintainability Symposium*, and will be published in the proceedings.

6 References

[1] P. S. Babcock. An introduction to reliability modeling of fault-tolerant systems. Technical Report CSDL-R-1899, C. S. Draper Laboratory, Inc., Cambridge, MA, September, 1986.

[2] R. E. Barlow and H. E. Lambert. *Introduction to Fault Tree Analysis*, pages 7–35. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1975.

[3] R. G. Bennetts. On the analysis of fault trees. *IEEE Transactions on Reliability*, R-24(3):194–203, August 1975.

[4] Andrea Bobbio and K. S. Trivedi. An aggregation technique for the transient analysis of stiff Markov chains. *IEEE Transactions on Computers*, C-35(9):803–814, September 1986.

* [5] M. A. Boyd, M. Veeraraghavan, Joanne Bechta Dugan, and K. S. Trivedi. An approach to solving large reliability models. In *AIAA/IEEE Digital Avionics Systems Conference, San Jose, CA*, October 1988.

[6] Mark A. Boyd. Converting fault trees to Markov chains for reliability prediction. Master's thesis, Duke University, Department of Computer Science, 1986.

[7] Joanne Bechta Dugan. *Extended Stochastic Petri Nets: Applications and Analysis.* PhD thesis, Department of Electrical Engineering, Duke University, 1984.

* [8] Joanne Bechta Dugan. Automated analysis of phased mission reliability. *IEEE Transactions on Reliability*, 1989. Submitted.

* [9] Joanne Bechta Dugan. Fault trees and imperfect coverage. *IEEE Transactions on Reliability*, June 1989.

* [10] Joanne Bechta Dugan, Salvatore Bavuso, and Mark Boyd. Fault trees and sequence dependencies. In *Proceedings of the Reliability and Maintainability Symposium*, 1989. To appear.

[11] Joanne Bechta Dugan, K. S. Trivedi, Mark K. Smotherman, and Robert M. Geist. The hybrid automated reliability predictor. *AIAA Journal of Guidance, Control and Dynamics*, 9(3):319–331, May-June 1986.

* [12] Joanne Bechta Dugan, Malathi Veeraraghaven, Mark Boyd, and Nitin Mittal. Bounded approximate reliability models for fault tolerant distributed systems. In *Proceedings 8th Symposium on Reliable Distributed Systems*, 1989.

[13] J. B. Fussell, E. F. Aber, and R. G. Rahl. On the quantitative analysis of priority-and failure logic. *IEEE Transactions on Reliability*, R-25(5):324–326, December 1976.

[14] J. B. Fussell and W. E. Vesely. A new methodology for obtaining cut sets for fault trees. *Transactions of the American Nuclear Society*, 15:262, 1972.

[15] A. Goyal, W. C. Carter, E. de Souza e Silva, S. S. Lavenberg, and K. S. Trivedi. The system availability estimator. In *Proceedings of the Sixteenth International Symposium on Fault-Tolerant Computing*, pages 84–89, July 1986.

[16] E. J. Henley and H. Kumamoto. *Reliability Engineering and Risk Assessment.* Prentice-Hall, 1981.

* acknowledges grant

[17] W. S. Lee, D. L. Grosh, F. A. Tillman, and C. H. Lie. Fault tree analysis, methods, and applications — a review. *IEEE Transactions on Reliability*, R-34(3):194–203, August, 1985.

[18] Mitchell O. Locks. Recursive disjoint products, inclusion-exclusion, and min-cut approximations. *IEEE Transactions on Reliability*, R-29(5):368–371, December 1980.

[19] A. Satyanarayana and A. Prabhakar. New topological formula and rapid algorithm for reliability analysis of complex networks. *IEEE Transactions on Reliability*, R-27(1):82–100, June 1978.

[20] D. P. Siewiorek and R. S. Swarz. *The Theory and Practice of Reliable System Design.* Digital Press, Bedford, MA, 1982.

[21] K. S. Trivedi. *Probability and Statistics with Reliability, Queueing and Computer Science Applications.* Prentice-Hall, Englewood Cliffs, NJ, 1982.